AD-A111 008 STATE UNIV OF NEW YORK AT BUFFALO AMHERST GROUP FOR --ETC F/G 12/2
TOWARD A THEORY OF STRATEGIES.(U)
DEC 81 N V FINDLER AFOSR-81-0220
UNCLASSIFIED GCSS-TR-6 AFOSR-TR-82-0048 NL

1 OF 1
AD A
111008

END
DATE
FILMED
03-82
DTIC

3

LEVEL II

12 29

TOWARD A THEORY OF STRATEGIES

BY

NICHOLAS V. FINDLER

DECEMBER 1981

DEPARTMENTAL TECHNICAL REPORT NUMBER: #194

GROUP FOR COMPUTER STUDIES OF STRATEGIES
TECHNICAL REPORT NUMBER: # 6

*THE WORK DESCRIBED HAS BEEN SUPPORTED BY THE AIR FORCE OFFICE OF SCIENTIFIC RESEARCH GRANT [redacted] AFOSR-81-0220

STATE UNIVERSITY OF NEW YORK AT BUFFALO

Department of Computer Science

412 787

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AFOSR-TR- 82 -0048

2. GOVT ACCESSION NO.
AD-A111008

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
TOWARD A THEORY OF STRATEGIES

5. TYPE OF REPORT & PERIOD COVERED
TECHNICAL

6. PERFORMING ORG. REPORT NUMBER
DEPT. TR 184; GCSS TR 6

7. AUTHOR(s)
Nicholas V. Findler

8. CONTRACT OR GRANT NUMBER(s)
AFOSR-81-0220

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Computer Science
State University of New York at Buffalo
4226 Ridge Lea Road, Amherst NY 14226

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61102F; 2304/A2

11. CONTROLLING OFFICE NAME AND ADDRESS
Mathematical & Information Sciences Directorate
Air Force Office of Scientific Research
Bolling AFB DC 20332

12. REPORT DATE
21 DEC 81

13. NUMBER OF PAGES
30

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Automatic analysis and synthesis of strategies, digital gaming, strategy vs. tactics, computer representation of static and learning strategies, credit assignment, descriptive and normative theories of strategies, the quasi-optimizer system, the advice taker/inquirer system, the generalized production rule system.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Some of the objectives and working tools of a new area of study, tentatively called Theory of Strategies, are described. It is based on the methodology of artificial intelligence, decision theory, utility theory, operations research and digital gaming. The latter refers to computing activity that incorporates model building, simulation and learning games in conflict situations.

Three long-term projects which aim at automatically analyzing and synthesizing strategies are also described.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TOWARD A THEORY OF STRATEGIES

Nicholas V. Findler
Department of Computer Science
Group for Computer Studies of Strategies
State University of New York at Buffalo

ABSTRACT

Some of the objectives and working tools of a new area of study, tentatively called *Theory of Strategies*, are described. It is based on the methodology of artificial intelligence, decision theory, utility theory, operations research and digital gaming. The latter refers to computing activity that incorporates model building, simulation and learning programs in conflict situations.

We also discuss three long-term projects which aim at automatically analyzing and synthesizing strategies.

1. INTRODUCTION

Mathematics has throughout the history of science served as both the queen and handmaiden of other disciplines in providing examples, stimulation and working tools for them. Computer Science has joined Mathematics in this role almost since computers appeared on the scene.

We describe a new offspring of established areas of study, which we tentatively call Theory of Strategies. It is characterized by the problems it aims at solving and by the methodology it would use.

At the start, it should be noted that the distinction between 'strategic' and 'tactical' is rather moot and varies from context to context. We shall refer to strategic considerations when their consequences remain relevant to the outcome of a confrontation throughout the conflict. A strategy is, of course, more than the sum of the participating tactics. It also includes the means of evaluating the adversary's situation and actions, scheduling of one's own tactics, and making use of feedback from the environment in modifying the rules of tactics both in terms of their contents and their inter-relations. In short, strategy gives tactics its mission and seeks to reap its results.

People use the word strategy in a variety of contexts. Although its original meaning ("the art of the general" in ancient Greek) refers to the conduct of warfare, the term has later assumed connotations ranging from statesmanship and management of national policy to diplomacy and economic planning, chiefly after the theoretical works by Karl von Clausewitz [1] and Antoine-Henri de Jomini [2]. After John von Neumann and Oskar Morgenstern [3] showed the similarity between the gamelike problems in economics, sociology, psychology and politics, the concept of strategy became pervasive also in social sciences. We talk about 'problem solving strategies' or the 'corporate strategy' in a large business enterprise, etc., whenever a sequence of goal-oriented actions is based on large-scale and long-range planning.

We shall adopt the latter type of interpretation of strategy and investigate how Computer Science can contribute to strategic planning.

2. THE OBJECTIVES OF A THEORY OF STRATEGIES

The following is a list of some of the objectives of the proposed theory:

.to identify adequate computer representations of static and learning strategies, which representations can then be effectively and efficiently employed both in a simulated world and in direct interaction with the real world;

.to develop techniques which analyze strategies, measure the performance of the whole strategy, and of its appropriately distinguished components ("credit assignment"), under most or all relevant conditions;

.to observe strategies in action--either in a sequence of unperturbed confrontations with others or under "laboratory conditions" when the environment is specified according to some experimental design--in order to generate a computer model (a "descriptive theory") of it;

.to combine the best components of several strategies, eliminate the redundancies and inconsistencies among these components and produce a strategy that is normative in the statistical sense;

.to establish stochastic, causal relationships between open variables that can be measured at any time and hidden variables whose values can be identified only intermittently or periodically, in order to find out the actions of a strategy, and their underlying reasons and consequences;

.to create a system that can be taught strategies via principles and high-level examples; the system should be able to make inquiries about vague, incomplete or contradictory advice, and to apply, evaluate and improve the strategy so acquired.

3. SOME METHODOLOGICAL ISSUES

The mathematical theory of games has given us a conceptual framework, a useful terminology but few practical methods to solve large-scale, complex, real-life problems.

On the other hand, areas of study such as decision theory, utility theory and operations research, could make important contributions to the technique we consider essential in approaching the above objectives.

We propose the term 'Digital Gaming' (DG) for the computing activity that incorporates model-building, simulation and learning programs. A programming system dedicated to DG, collaborating with human decision-makers, would eventually assume the role of a Command-and-Control unit.

The idea of attacking the problem of strategic planning with the techniques of Computer Science has several "by-products" that we note here. As any person in computing would tell, when one has to formulate a problem to program it, all "intangibles" must be described so as to be amenable to algorithmic or heuristic treatment. Such description also clarifies the thought processes of the experts whose advice and experience are sought in establishing the programming system. Thus even the existing techniques are bound to improve.

More importantly, the designer of a system, working in the top-down mode, assumes the existence of modules below the one he is concerned with. He establishes a flow of control and information among subsystems which will be implemented later or, possibly, are to be operated by human beings for some time to come. This idea should encourage continual expansion of the domain already automated.

4. ON DIGITAL GAMING

As noted before, DG is more than running simulation models. We believe in the utility of machine learning, which has been in the focus of our interest in studying decision-making under uncertainty and risk [4-8]. Learning programs would assume an important role in DG [11]. They would continually improve the performance of the system whenever (i) better responses are attainable under constant environmental conditions, (ii) the physical environment or the adversary strategy changes. In order to illustrate their relevance, we describe briefly three types of learning processes (out of some two dozen) and a high-level strategy-acquisition technique that we have been working on over the past several years.

5. THE "BAYESIAN" LEARNING MODE

"Bayesian" learning processes make inductive inferences, that is, draw general conclusions from specific events. (The name refers to Bayes' theorem in probability which assumes the a priori knowledge of certain conditional probabilities of certain events occurring after some other events.) They modify the decision-making rules by comparing predicted outcomes of events and actual outcomes. There are basically three ways to adjust the rules to bring the actual outcomes closer to the expected ones. If a number of parameters are included in the pre-established heuristic rules, a learning process can make their values converge to near-optimum values. Or an optimum hierarchical ordering for the heuristic rules can be found experimentally. It is also possible to generate automatically new heuristic rules, test them and incorporate the successful ones into the new strategy--a usually difficult and time-consuming process.

It should be noted that, in accordance with the experimental spirit of DG, a variety of "Bayesian" learning processes must be tried, which vary in the type and amount of information they collect and in how they use it.

6. THE QUASI-OPTIMIZER (QO) SYSTEM

Let us consider an environment in which either several organizations are competing to achieve an identical, mutually conflicting goal, or else a set of alternative strategies exist, each trying to win against an identical, opposing strategy [9,13]. (One can assume, for the sake of generality, that a goal vector is specified whose components need not be independent in real-life confrontations; for example, in air battle management, the ratio of targets accessed and enemy air defense units suppressed are obviously inter-related goal components.)

Each of the strategies evaluates the environment by measuring certain variables (numerical or symbolic) available to it, which the strategy considers relevant. Such variables may be the real or assumed actions of the adversary, the perceived state of the confrontation, availability and capabilities of friendly forces, threat estimates, criticality and vulnerability of the adversary's and our resources, etc. An important component of a strategy is aimed at interpreting these measurements and incorporating them in the process of making decisions that can lead to goal-achievement (and to the exclusion of goal-achievement by the adversary).

The environment as perceived by the strategy is unclear because some information may be unavailable, missing (risky or uncertain, according to whether or not the relevant a priori probability distributions are known, respectively) or obscured by noise (caused accidentally or by deliberate obfuscation). If the decisions based on such incomplete and/or inconsistent information are less sound than those of the adversary, resources will be wasted and goal achievement will be farther removed.

Let us now consider how we could generate a new strategy. The system has to generate automatically a model (a descriptive theory) of every participating strategy through observation and measurements. It would then have to assign to each component of the models some measure of quality; that is, an outcome-dependent allocation of credit must be made.

The strategy obtainable from the best components of the model strategies is a normative theory which is potentially the best of all available ones, on the basis of the information accessible by us. This normative strategy is in fact only quasi-optimum for four reasons. First, the resulting strategy is optimum only against the original set of strategies considered. Another set may well employ controllers and indicators for decision-making that are superior to any in the "training" set. Second, the strategy is normative only in the statistical sense. Fluctuations in the adversary strategy, whether accidental or deliberate, impair the performance of the QO strategy. Third, the adversary strategy may change over time and some aspects of its dynamic behavior may necessitate a change in the QO strategy. Finally, the generation of both the descriptive theories (models) and of the normative theory (the QO theory) is based on approximate and fallible measurements.

The system under development employs the following modules:

6.1 The QO-1 assumes a monotonic strategy response surface and uses either exhaustive search or binary chopping to construct a descriptive theory of static (non-learning) strategies.

6.2 The QO-2 extrapolates a finite sequence of learning trees, each representing the same strategy at different stages of development, and computes their asymptotic form.

The latter will then be used in constructing the normative theory.

6.3 The QO-3 minimizes the total number of experiments QO-1 has to perform. It no longer assumes that the strategy response surface is monotonic and will eventually also deal with multi-dimensional responses. QO-3 starts with a balanced incomplete block design for experiments and computes dynamically the specifications for each subsequent experiment. In other words, the levels of the decision variables in any single experiment and the length of the sequence of experiments depend on the responses obtained in previous experiments.

6.4 The QO-4 performs the credit assignment. That is, it identifies the components of a strategy and assigns to each a quality measure of the 'outcomes'. An outcome need not be only the immediate result of a sequence of actions prescribed by the strategy but can also involve long-range consequences of planned actions.

6.5 The QO-5 constructs a 'Super Strategy' by combining strategy components associated with outcomes of a quality above a threshold value.

6.6 The QO-6 generates a Quasi-Optimum strategy from the Super Strategy by eliminating inconsistencies and redundancies from the latter. It also tests and verifies the QO strategy for completeness.

7. THE ADVICE TAKER/INQUIRER SYTEM (AT/I)

The objective of this system is to establish a man-machine environment in which a human advisor can teach strategies of confrontation on-line, through principles and high-level examples. The principles and examples normally consist of situations and recommended actions. (Principles describe rather general situations defined in a flexible manner whereas examples are specific and illustrate appropriate behavior in a general situation by analogy with a particular one. Actions can either adhere to some general guidelines or follow a set of sharply defined prescriptions.) Whenever the system finds the advice given to be vague, incomplete or inconsistent with previously imparted knowledge, it makes inquiries and asks for clarification. The advisor can define and re-define the components of a principle at any time. He can also over-ride temporarily the strategy taught so far by issuing a order.

The system does not start out with a blank memory. It knows the rules governing the confrontation, the variables, and the ranges of their values within the situation space. The advisor can at any time

(i) define variables, functions, general and specific actions, confrontation-related adjectives, nouns and verbs--in terms of constants, confrontation parameters, current values, overall and moving averages of statistical values, basic confrontation actions, and

Boolean and relational operators;

(ii) define principles of a strategy which connect a situation (specified as a Boolean combination of ranges of statistical variables--again current values, overall or moving averages) to some general or specific action;

(iii) give high-level examples by connecting sharply specified situations to direct confrontation actions;

(iv) make inquiries about definitions, principles, and values of statistical variables stored so far;

(v) issue an order which temporarily over-rides the strategy acquired so far.

In turn, the system can

(a) ask for clarification whenever new definitions are vague or conflict with stored ones, or the strategy is incomplete in not covering the whole confrontation space;

(b) return exemplary actions in user-specified confrontation situations, in accordance with the strategy acquired;

(c) display definitions, principles, confrontation parameters, values of variables, etc.

Random number generators also have a role in defining game-theoretically mixed strategies. A sense of time has also to be incorporated in the "tool kit" of definitions, whether it refers to continuous or quantized time or to a counter of certain specific events.

We note two important facilities to be used in specifying principles. Let us call these Advisor-Assigned and Advisor-Defined Adversary Types (AAAT and ADAT, respectively). In the former case, the advisor assigns a certain adversary to one or more categories (Adversary Types) named by him. In the latter case, the advisor defines one or several categories by Boolean combinations of ranges of statistical variables, which are regularly or continually collected over the adversary's actions. (The variables can refer to current values, or overall or moving averages.) At prescribed intervals, the system compares the adversary behavior with the specifications of all ADAT's. Accordingly, each adversary (at that time) may belong to various Advisor-Defined Adversary Types. Thus the principle describing the appropriate action can refer to all such adversaries that satisfy the definition conditions of the Adversary Type at hand.

Advisor-defined nouns can reasonably be required to be unambiguous. However, adjectives (and, to some extent, verbs) must often have different meanings when used to modify different types of nouns (cf. a "strong attack" vs. a "strong concentration"). The AT/I system has to distinguish (at least) four different classes of instances:

(i) Patent: confrontation parameters, statistical variables, AT/I's own resources (e.g., "If your air superiority is more than 2:1, seek air battles.")

(ii) Interactive: the adversary's actions during current confrontation (e.g., "If the adversary is bringing up additional resources, assume a holding position.")

(iii) Statistical: accumulated data about the adversary's past behavior (e.g., "If the adversary is self-confident, make sudden attacks.")

(iv) Inferential: assumptions about the intentions or events behind the adversary's behavior (e.g., "If the enemy appears to have received additional supplies, wait for confirmation.")

This classification is neither exhaustive nor exclusive. If the Definition Manager, a part of the programming system, cannot decide unambiguously on the class into which the components of the definition fall, it has to consult the human advisor.

Another difficulty rests with the need to resolve a situation-dependent conflict between principles of global and monetary relevance. Futhermore, the system must be able to generate disambiguating questions whenever the relative importance of the principles, as specified by the advisor, is inconsistent because of non-transitive preferences given in the advice.

Finally, we note that to teach a strategy by telling *how* to do things in general is more efficient and less error-prone than to tell *what* to do in every relevant situation. An AT/I-like system would have practical usefulness in doing this. Human experts would specify, via a high-level interaction with the machine, a number of alternative strategies. Other components of the DG system, such as a QO-like system, would then generate models of uniform structure of each strategy. A prescriptive, quasi-optimum strategy would then finally be constructed from these.

The system under construction employs the following modules:

7.1 The *AT/I-1* constructs the framework for the flow of information and control between the AT/I system and the Advisor.

7.2 The *AT/I-2* converts the principles and high-level examples into a canonical form and stores them. Next it embeds them into an initially skeleton strategy which then becomes employable.

7.3 The *AT/I-3* eliminates inconsistencies and incompletenesses from the strategy acquired, in part by interacting with the Advisor.

7.4 The *AT/I-4* tests (verifies) and evaluates the strategy constructed according to a metric which is independent of any particular strategy.

5. THE GENERALIZED PRODUCTION RULES (GPR) SYSTEM

The underlying motivations for the actions prescribed by a strategy, the actions themselves, and their consequences are not necessarily observable and measurable at any desired time. The values of such *hidden variables* can be identified only at certain times, either intermittently or periodically. At other times, their values have to be estimated. In contrast, the *open variables* are readily measurable at any time. The estimation is based on generalized production rules expressing stochastic, causal relations between open and hidden variables. Either can be cause or effect. The GPR system is designed to provide decision support for expert systems in need for numerical estimates of hidden variable values.

A knowledge base is established over a period of measurements. It consists of an ordered set of generalized production rules of the form

$$W_r / M_{ijk} / T_{jn} \Rightarrow V_m (H_n) \quad : C_r$$

Here $\underline{W_r}$ is the number of rules that have been pooled to form the $\underline{r}$-th rule. $\underline{M_{ijk}}$ is the $\underline{i}$-th combination of the parameters of the $\underline{j}$-th basic pattern (morph) describing the behavior of the *k*-th open variable (OV). T_{jm} is the difference in time (timelag) or in space (distance) between the start of the *j*-th morph (in case of a trend) or its occurrence (in case of a sudden change or step function), and the point of time or space at which the *n*-th HV, H_n, assumes its *m*-th value, V_m. This difference may be positive--when the OV is the cause and thus precedes the HV, the effect--or negative in the opposite case. The term 'lag' is used for T_{jm}, whether it refers to a timelag or distance. Q_r is the credibility level of the *r*-th rule. Its value is between 0 and 1, and depends on two factors:

.how well the morph in question fits the datapoints over its domain, and

.how many and how similar the rules were that have been pooled to form the rule at hand.

When an estimate of a HV value is desired at a certain value of the lag variable, the user has to provide in its vicinity a sequence of values of all available OV's that are assumed to be causally related to the HV. These sequences are then submitted to the morph-fitting program (MFP). The system then looks in the knowledge base for the *N* best estimates (*N* specified by the user) coming from rules that

.connect the HV sought and the available OV's;

.refer to the same type of morph as the newly fitted one;

.involve morph parameters and lag values that are "similar enough" to those in the query, i.e. that are within the user-specified range of pooling rules.

The so-called confidence level of the estimate, C_e, depends on the credibility level of the rule used as well as how well the new morph fits its datapoints and how close its parameters are to those of the morph matched in the knowledge base.

Let us now assume that the estimation is performed and up to N values of the HV are returned for each lag value that yields such possibility. The system will calculate the average of the N estimates weighted by their confidence level. This process thus provides datapoints, each specifying weighted average HV vs. lag value, over the whole range of interest. The system then finally invokes the MFP to produce the functional form desired. Its validity is based on the assumption that the OV's, whose morphs were used for the estimation, have obeyed the same laws when the observations were made for the knowledge base as when they were measured for the estimation. Furthermore, the relations between and within the groups of OV's and HV's are, statistically speaking, constant over time.

The system employs the following modules:

8.1 The GPR-1 fits a minimal set of basic patterns, morphs, to a sequence of open variable datapoints.

8.2 The GPR-2 establishes rules between sets of parametric values of morphs describing open variable behavior and individual values of hidden variables.

8.3 The GPR-3 pools rules that connect the same open variable and hidden variable and satisfy certain statistical and rule-generation criteria. The number and credibility of rules increase with experience.

8.4 The GPR-4 estimates the values of hidden variables at desired time points.

8.5 The GPR-5 extends the system to distributed processing and intelligence. It merges source files and knowledge bases, established at different observation points by satellite computers, if certain statistical and file-generation criteria are satisfied--as verified by the system automatically.

8.6 The GPR-6 extends the system's capabilities to estimating the functional form of hidden variable distributions rather than estimating only the individual values of hidden variables.

9. AIR TRAFFIC CONTROL (ATC) AS A TASK ENVIRONMENT FOR THE THEORY OF STRATEGIES

Students of all emerging disciplines soon feel the need to employ their newly developed tools on some real-life problems. The recent shift toward applicable research in Artificial Intelligence clearly indicates that this area of study has matured sufficiently. Production systems incorporate extensive bases of expert knowledge in a variety of different domains. Event-driven process models can simulate realistic, large subsets of the real world. Problem-solving techniques have become powerful enough to control complex robot behavior. The ATC environment seems to have the following important qualifications for being studied within the technical and conceptual framework of the Theory of Strategies:

.the task is complex enough to be challenging;

.one can identify problem areas of different sizes that could be attacked successively;

.one can define plausible metrics along several dimensions to measure the performance of a proposed system;

.until a subsystem is fully developed and tested, it can operate in a realistic, simulated world;

.a successful system for automatic ATC would share with systems working in other environments the important capabilities of planning, problem-solving and decision-making under uncertainty and risk in dynamically changing domains while satisfying a hierarchy of constraints.

Interest in automating the ATC task has increased over the past few years [19-25]. The need for radical modernization of the current mode of operation, as shown by the number of near-misses mostly due to errors in human judgement, has been made more critical by the recent controversy between the Federal Government and PATCO.

In the following, we intend to discuss the above issues briefly, outline an "ideal" ATC system, and show how our present work could contribute to the development of automated ATC.

When an aircraft flies from one airport to another under instrument flight rules (as military and civilian planes do), it passes through the jurisdiction of a series of ATC centers. These centers track each flight within their sector on radar and try to keep it on its appointed path, according to a desired time schedule. The control actions must also satisfy a number of constraints. Some are constant, such as the government-prescribed rules for minimum separation and the physical limitations of aircraft capabilities. Others arise from the situation, such as unfavorable weather conditions and emergency landing priorities. In addition to safe and timely take-offs, flights and landings, fuel economy and noise pollution over inhabited areas must also be considered.

The above microcosmos is well-structured in terms of state changes over space and time. The commands and pilot actions are drawn from a small standardized set. The measures of aircraft performance are simple, such as flight time, fuel consumed, and number and degree of constraint violations. Systems competence can be measured along the dimensions of the number and the duration of validity of commands, and (assuming perfect adherence to the commands)

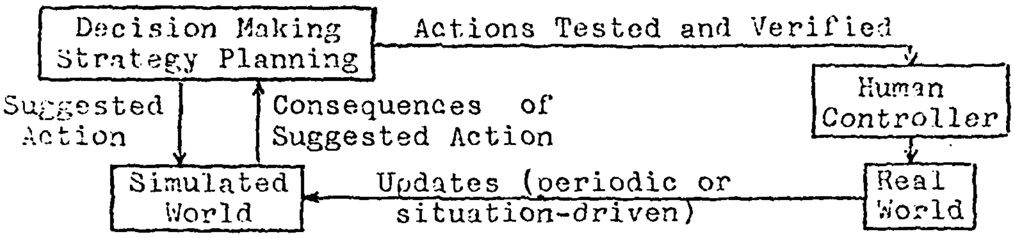
Decision Making
Strategy Planning
Actions Tested and Verified
Human
Controller
Suggested
Action
Consequences of
Suggested Action
Simulated
World
Updates (periodic or
situation-driven)
Real
World

FIGURE 1

all the measures of aircraft performance.

Sources of uncertainty are due to imperfect adherence to commands, fuzziness in location of aircraft on radar images, suboptimal commands issued, unexpected environmental events/weather, incoming aircraft, etc.)

Figure 1 shows an idealized arrangement for an automated ATC system. The strategy, based on plans, is tested and verified in the Simulated World. The consequences of the actions suggested are fed back to the Decision-Making unit and, if the results are unsatisfactory, the actions are modified as long as necessary.

The actions thus proven are then communicated by a human controller as a command to the Real World. Finally, the status of the Real World updates that of the Simulated World at regular intervals or more often in critical situations.

FIGURE 1 ABOUT HERE

We shall show how our present work can contribute to the ATC task. Assume that ATC trainees specify their control strategy in terms of principles and high-level examples to the Advice Taker/Inquirer (AT/I) system. The latter is linked up with the Simulated World in which it tests, verifies and evaluates the consequences of the strategy so imparted. It seems from the educational point of view, therefore, a very useful feedback loop that involves the ATC trainee, the AT/I system and the Simulated World.

The Quasi-Optimizer (QO) system would automatically generate a computer model, a descriptive theory, of the trainees' strategies. Finally, it would create a normative theory, quasi-optimum for reasons described before, out of the descriptive theories.

In view of the well-defined boundaries of this problem-solving universe and of the limited set of distinct situations and actions, it is likely that our theoretical efforts can be employed for this important, practical domain.

10. FINAL COMMENTS

We have introduced a Theory of Strategies in terms of its objectives and some possible techniques and methodologies. We have shown approaches to automatic analysis and synthesis of strategies. We have introduced the term term Digital Gaming, which involves model-building, simulation and machine learning ideas. Digital Gaming, augmented with the tools of operations research, decision theory and utility theory, would provide a computational environment to automate important aspects of strategic planning.

11. ACKNOWLEDGEMENTS

The work described above was or is being done by our Group for Computer Studies of Strategies. I am indebted to each of its past and present members. In particular, George L. Sicherman has been a constant discussion partner and critic. Michael Belofsky did the text processing of this paper. We have been supported by the Air Force Office of Scientific Research Grant 81-0220.

12. REFERENCE

[1] Clausewitz, K. von: *On War*(Transl. by J. J. Graham; Routledge & Kegan Paul, Ltd.: London, 1940).

[2] Jomoni, A.-H. de: *Summary of the Art of War*(Transl. by O. F. Winship & E. E. Mc Lean; G. P. Putnam: New York, 1854).

[3] Neumann, John von: *Theory of Games and Economic Behavior*(Third ed., Princeton Univ. Press: Princeton, 1953).

[4] Findler, N. V., H. Klein, W. Gould, A. Kowal, and J. Menig: Studies on decision making using the game of Poker (*Proc. IFIP Congress 71*, Ljubljana, Yugoslavia, pp. 1448-1459; North-Holland: Amsterdam 1972).

[5] Findler, N. V., H. Klein, and Z. Levine: Experiments with inductive discovery processes leading to heuristics in a Poker program (In Beckmann, Goos, and Künzi (Eds.): *Cognitive Processes and Systems*, Springer: Berlin, 1973).

[6] Findler, N. V., H. Klein, R. C. Johnson, A. Kowal, Z.

Levine and J. Henig: Heuristic programmers and their gambling machines (Proc. ACM Nat. Conf., San Diego, CA, pp. 26-37, 1974).

[7] Findler, N. V.: Studies in machine cognition using the game of Poker (Comm. ACM, 20, pp. 230-245,1977).

[8] Findler, N. V.: Computer Poker (Scientific American, pp. 148-151, July 1979).

[9] Findler, N. V. and J. van Leeuwen: The complexity of decision trees, the Quasi-Optimizer, and the power of hueristic rules (Information and Control, 40, pp. 1-19, 1979).

[10] Findler, N. V., G. L. Sicherman, and S. Feuerstein: Teaching strategies to an Advice Taker/Inquirer system (Proc. Euro IFIP79 Conf., London, England, pp. 457-465,1979).

[11] Findler, N. V.: Aspects of computer learning (Cybernetics and Systems, 11, pp. 65-84, 1980).

[12] Findler, N. V.: Pattern recognition and generalized production systems in strategy development (Proc. Fifth Internat. Conf. on Pattern Recognition; Miami, Fla. Vol. I, pp. 140-150, 1980).

[13] Findler, N. V. and J. P. Martins: On automating computer model construction--the second step toward a Quasi-Optimizer system (Journal of Information and Optimizing Sciences, 2, pp. 119-136, 1980).

[14] Findler, N. V.: A multi-level learning technique using

production systems (Cybernetics and Systems, to appear in March 1982).

[15] Findler, N. V. and E. Morgado: Morph-fitting--An effective technique of approximation (IEEE Trans. on Pattern Analysis and Machine Intelligence, Special Issue: Computer Intelligence--Three Decades, to appear in March 1982).

[16] Findler, N. V.: An expert subsystem based on generalized production rules (Submitted for publication).

[17] Findler, N. V., H. Mazur, and B. McCall: A note on computing the asymptotic form of a limited sequence of decision trees (Submitted for publication).

[18] Findler, N. V., J. E. Brown, R. Lo, H. Y. You: A module to estimate numerical values of hidden variables for expert systems (Submitted for publication).

[19] Air Traffic Control (Federal Aviation Administration, Department of Transportation, January 1, 1980).

[20] Conker, R.S.: An overview description of vertical profile modeling in the AERA test bed (Build 1) (MITRE Working Paper, WP-80W00702, Sep. 1980).

[21] Conker, R.S. and J.A. Kingsbury : Description of the clearance directives for Build 2 of the AERA test bed (MITRE Working Paper, WP-81W00095, Feb. 1981).

[22] Swetnam, G.F., Jr. : A preliminary characterization of the AERA man-machine interface for ATC computer replacement (MITRE Working Paper, WP-81W00135, March 1981).

[23] Goldmuntz, L., J.T. Kefaliotis, L.A. Kleiman, R.A.

Rucker, L. Schuchman, and D. Weathers : Automated en-route air traffic control (AERA) concept (Federal Aviation Authority, Department of Transportation, March 1981).

[24] Wesson, R.B. : Planning in the world of the air traffic controller (Proc. IJCAI-77, Cambridge, MA; pp. 473-479, 1977).

[25] Wesson, R.B. : Problem-solving with simulation in the world of an air traffic controller (Unpublished Ph.D. dissertation, University of Texas, Austin, TX, 1977).

DATE
FILMED

3-8